



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

## Multilevel quasiseparable matrices in PDE-constrained optimization

### Citation for published version:

Zhlobich, P & Gondzio, J 2011 'Multilevel quasiseparable matrices in PDE-constrained optimization' ArXiv.  
<<http://arxiv.org/abs/1112.6018>>

### Link:

[Link to publication record in Edinburgh Research Explorer](#)

### Document Version:

Early version, also known as pre-print

### General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

### Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact [openaccess@ed.ac.uk](mailto:openaccess@ed.ac.uk) providing details, and we will remove access to the work immediately and investigate your claim.



# Multilevel quasiseparable matrices in PDE-constrained optimization

Jacek Gondzio · Pavel Zhlobich

the date of receipt and acceptance should be inserted later

**Abstract** Optimization problems with constraints in the form of a partial differential equation arise frequently in the process of engineering design. The discretization of PDE-constrained optimization problems results in large-scale linear systems of saddle-point type. In this paper we propose and develop a novel approach to solving such systems by exploiting so-called *quasiseparable* matrices. One may think of a usual quasiseparable matrix as of a discrete analog of the Green's function of a one-dimensional differential operator. Nice feature of such matrices is that almost every algorithm which employs them has linear complexity. We extend the application of quasiseparable matrices to problems in higher dimensions. Namely, we construct a class of preconditioners which can be computed and applied at a linear computational cost. Their use with appropriate Krylov methods leads to algorithms of nearly linear complexity.

**Keywords** saddle-point problems, PDE-constrained optimization, preconditioning, optimal control, linear systems, quasiseparable matrices

**Mathematics Subject Classification (2010)** 49M25 · 49K20 · 65F08 · 65F10 · 65F50 · 65N22

## 1 Introduction

In this paper we introduce a new class of structured matrices that we suggest to call *multilevel quasiseparable*. It generalizes well-known quasiseparable matrices to the multi-dimensional case. Under the multilevel paradigm, parameters that are used to represent a matrix of a higher hierarchy are themselves multilevel quasiseparable of a lower hierarchy. The usual one-dimensional quasiseparable matrix is the one of the lowest hierarchy.

---

School of Mathematics, The University of Edinburgh, JCMB, The King's Buildings, Edinburgh, Scotland EH9 3JZ, E-mail: J.Gondzio@ed.ac.uk, P.Zhlobich@ed.ac.uk.

Quasiseparable matrices found their application in orthogonal polynomials [1], root finding [4], system theory [2] and other branches of applied mathematics and engineering. Consequently, there has been major interest in them in the scientific community, and many fast algorithms for working with them have been developed in the last decade. Due to the similarity in representations of multilevel quasiseparable and usual quasiseparable matrices, it is possible to extend the applicability of almost all the algorithms developed for quasiseparable matrices to the new class. It is a celebrated fact that operations with quasiseparable matrices can be performed in linear time with respect to their size. We show that under some conditions this is also true for multilevel quasiseparable matrices.

Natural applications of multilevel quasiseparable matrices are discretized Partial Differential Equations (PDEs) and related problems. There have already been recent attempts to use structured matrices similar to quasiseparable ones to solve discretized PDEs [25, 19]. The very distinct feature of our approach is the ability not only to solve systems in linear time, but also to *invert* matrices and to perform arithmetic operations with them in linear time. Therefore, we can solve, for instance, saddle-point systems with PDE matrices that former approaches did not allow.

The structure of the paper is as follows: we start with a motivation for this paper by considering a simple PDE-constrained optimization problem in Section 2. This type of problems leads to systems in a saddle-point form. The need in efficient solvers for such systems motivated this paper. In Section 3 we briefly review quasiseparable matrices and related results. Section 4 concerns the extension of quasiseparable matrices to the multi-dimensional case. We give there a formal definition of multilevel quasiseparable matrices, introduce their parametrization and fast arithmetic with them. The last section presents results of preliminary numerical experiments that empirically confirm linear time complexity of algorithms with multilevel quasiseparable matrices.

## 2 Model PDE-constrained optimization problem.

As a simple model let us consider a distributed control problem which is composed of a cost functional

$$\min_{u, f} \frac{1}{2} \|u - \hat{u}\|_2^2 + \beta \|f\|^2 \quad (1)$$

to be minimized subject to a Poisson's problem with Dirichlet boundary conditions:

$$\begin{aligned} -\nabla^2 u &= f \text{ in } \Omega, \\ u &= g \text{ on } \partial\Omega. \end{aligned} \quad (2)$$

The problem consists in finding a function  $u$  that satisfies the PDE constraint and is as close as possible in the L2 norm sense to a given function  $\hat{u}$  ("desired state"). The right-hand side of the PDE  $f$  is not fixed and can be designed in order to achieve the optimality of  $u$ . In general such a problem

is ill-posed and thus needs Tikhonov regularization that is introduced in the form of the second term in (1).

The usual approach to solving the minimization problem (1) – (2) numerically is to introduce a weak formulation of it and to discretize the latter by finite elements, see, for instance, [21, 12]. After these steps, the discrete analog of (1) – (2) reads

$$\begin{aligned} \min_{u_h, f_h} \quad & \frac{1}{2} \|u_h - \widehat{u}\|_2^2 + \beta \|f_h\|_2^2, \quad u_h \in V_g^h, f_h \in V_0^h, \\ \text{subject to} \quad & \int_{\Omega} \nabla u_h \cdot \nabla v_h = \int_{\Omega} v_h f, \quad \forall v_h \in V_0^h, \end{aligned} \quad (3)$$

where  $V_0^h$  and  $V_g^h$  are finite-dimensional vector spaces spanned by test functions. If  $\{\phi_1, \dots, \phi_n\}$  and  $\{\phi_1, \dots, \phi_n, \phi_{n+1}, \dots, \phi_{n+\partial n}\}$  are bases in  $V_0^h$  and  $V_g^h$ , respectively, and  $u_h$  and  $v_h$  are represented in them as

$$u_h = \sum_{j=1}^{n+\partial n} u_j \phi_j, \quad f_h = \sum_{j=1}^n f_j \phi_j,$$

then the matrix form of (3) is

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{f}} \quad & \frac{1}{2} \mathbf{u}^T M \mathbf{u} - \mathbf{u}^T \mathbf{b} + c + \beta \mathbf{f}^T M \mathbf{f}, \\ \text{subject to} \quad & K \mathbf{u} = M \mathbf{f} + \mathbf{d}, \end{aligned} \quad (4)$$

where  $M_{ij} = \int_{\Omega} \phi_i \phi_j$  — mass matrix,  $K_{ij} = \int_{\Omega} \nabla \phi_i \nabla \phi_j$  — stiffness matrix,  $b_j = \int_{\Omega} \phi_j \widehat{u}$ ,  $c = \frac{1}{2} \int_{\Omega} \widehat{u}^2$ ,  $d_j = \sum_{k=n+1}^{n+\partial n} u_k \int_{\Omega} \nabla \phi_k \nabla \phi_j$ ,  $i, j = 1, \dots, n$ .

The Lagrangian associated with problem (4) is

$$\mathcal{L}(\mathbf{u}, \mathbf{f}, \boldsymbol{\lambda}) = \frac{1}{2} \mathbf{u}^T M \mathbf{u} - \mathbf{u}^T \mathbf{b} + c + \beta \mathbf{f}^T M \mathbf{f} + \boldsymbol{\lambda}^T (K \mathbf{u} - M \mathbf{f} - \mathbf{d}),$$

where  $\boldsymbol{\lambda}$  is a vector of Lagrange multipliers. The condition for a stationary point of  $\mathcal{L}$  define  $\mathbf{u}$ ,  $\mathbf{f}$  and  $\boldsymbol{\lambda}$  via the solution of the linear system

$$\begin{bmatrix} 2\beta M & 0 & -M \\ 0 & M & K^T \\ -M & K & 0 \end{bmatrix} \begin{bmatrix} \mathbf{f} \\ \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \\ \mathbf{d} \end{bmatrix}. \quad (5)$$

This linear system is of saddle-point type. We refer to [3] for an extensive survey of numerical methods for this type of systems.

Matrix in (5) is symmetric, usually very large but sparse due to finite element discretization. Thus matrix-vector multiplication is cheap and Krylov subspace methods such as Minimal Residual (MINRES) method of Paige and Saunders [20] or Projected Preconditioned Conjugate Gradient (PPCG) method of Gould, Hribar and Nocedal [15] are well suited. Their convergence nevertheless depends highly on the choice of a good preconditioner [21].

In this paper we propose a structured matrices approach to the solution of systems of the type described above. First, let us write the block LDU decomposition of (5):

$$\begin{bmatrix} 2\beta M & 0 & -M \\ 0 & M & K^T \\ -M & K & 0 \end{bmatrix} = \begin{bmatrix} I & 0 & 0 \\ 0 & I & 0 \\ \frac{-1}{2\beta}I & KM^{-1} & I \end{bmatrix} \cdot \begin{bmatrix} 2\beta M & 0 & 0 \\ 0 & M & 0 \\ 0 & 0 & S \end{bmatrix} \cdot \begin{bmatrix} I & 0 & \frac{-1}{2\beta}I \\ 0 & I & M^{-1}K^T \\ 0 & 0 & I \end{bmatrix}, \quad (6)$$

$$S = - \left( \frac{1}{2\beta}M + KM^{-1}K^T \right).$$

The hardest part in using this decomposition for solving a system is to compute the Schur complement  $S$  of the (3,3) block and to solve the corresponding system with it. Since matrix  $S$  is usually dense, this task is untractable if we use its entrywise representation but it is feasible if we use a proper structured representation. In Section 4 we will show that Schur complement indeed has a structure and this structure is the one called *multilevel quasiseparable*. We will give all necessary definitions and show that the use of multilevel quasiseparable matrices leads to asymptotically (with respect to the size of the system) linear storage and linear complexity algorithm for solving (5).

### 3 Quasiseparable matrices.

Matrix of a rank much smaller than its size is a discrete analog of a separable function. More generally, we may think of a matrix with certain blocks being low rank rather than the whole matrix. This case corresponds to a function being separable in some subdomains of the whole domain. One simple example of such a function is Green's function of the Sturm–Liouville equation (this example will be considered in some more details later in this section). There is no surprise that matrices with low rank blocks found their applications in many branches of applied mathematics, especially in integral equations. There are several related classes of rank-structured matrices that one can find in the scientific literature. Among the most well-known are semiseparable [14], quasiseparable [9],  $H$ -matrices [16],  $H^2$ -matrices [17] and mosaic-skeleton matrices [23]. In the current paper our main tool will be quasiseparable matrices [24] having low rank partitions in their upper and lower parts. The formal definition is given below.

**Definition 1 (Rank definition of a block quasiseparable matrix)** Let  $A$  be a block matrix of block sizes  $\{n_k\}_{k=1}^n$  then it is called block  $(r^l, r^u)$ -quasiseparable if

$$\max_K \text{rank } A(K+1 : N, 1 : K) \leq r^l, \quad \max_K \text{rank } A(1 : K, K+1 : N) \leq r^u,$$

$$K = \sum_{i=1}^k n_i, \quad N = \sum_{i=1}^n n_i,$$

where  $r^l$  ( $r^u$ ) is called lower (upper) order of quasiseparability.

In what follows we will call *block*  $(r^l, r^u)$ -*quasiseparable* matrices simply *quasiseparable* for shortness.

A well-known example of a quasiseparable matrix is given by the discretized Green's function of a one-dimensional differential equation. Consider a regular inhomogeneous Sturm-Liouville boundary value problem:

$$\begin{aligned} (p(x)u')' - q(x)u &= f(x), \\ \begin{cases} \alpha_1 u(a) + \beta_1 u'(a) = 0, \\ \alpha_2 u(b) + \beta_2 u'(b) = 0, \end{cases} \end{aligned} \quad (7)$$

where functions  $p, p', q$  are continuous on  $[a, b]$ ,  $p(x) > 0$  on  $[a, b]$  and  $|\alpha_i| + |\beta_i| \neq 0$  for  $i = 1, 2$ . It is a classical result in ordinary differential equations that the solution  $f(x)$  of (7) is given by

$$u(x) = \int_a^b g(x, \xi) f(\xi) d\xi,$$

where  $g(x, \xi)$  is the so-called Green's function. In this case it has an explicit formula

$$g(x, \xi) = \frac{1}{p(a)W(a)} = \begin{cases} u_1(x)u_2(\xi), & a \leq x \leq \xi, \\ u_1(\xi)u_2(x), & \xi < x \leq b, \end{cases} \quad (8)$$

with  $W(x)$  being Wronskian and  $u_1(x)$  and  $u_2(x)$  being two specific linearly independent solutions of (7). It is easy to see from (8) that discretized Green's function is a quasiseparable matrix of order one.

In order to exploit the quasiseparability of matrices in practice one must have a low-parametric representation of them. There are many alternative parametrisations of quasiseparable matrices all of which use  $\mathcal{O}(N)$  parameters, where  $N$  is the size of the matrix. Having such a parametrisation at hand one can write most of the algorithms, e.g., inversion, LU or QR decomposition, matrix-vector multiplication in terms of these parameters and, therefore, the complexity of these algorithms is  $\mathcal{O}(N)$ . In this paper we will use the so-called *generator representation* (Definition 2 below) proposed by Eidelman and Gohberg [9]. There are several alternative parametrisations such as Givens-weight [7] and Hierarchical Semiseparable (HSS) [6].

**Definition 2 (Generator definition of a block quasiseparable matrix)**

Let  $A$  be a block matrix of block sizes  $\{n_k\}_{k=1}^n$  then it is called block  $(r^l, r^u)$ -quasiseparable if it can be represented in the form

$$\begin{bmatrix} d_1 & g_1 h_2 & g_1 b_2 h_3 & \cdots & g_1 b_2 \cdots b_{n-1} h_n \\ p_2 q_1 & d_2 & g_2 h_3 & \cdots & g_2 b_3 \cdots b_{n-1} h_n \\ p_3 a_2 q_1 & p_3 q_2 & d_3 & \cdots & g_3 b_4 \cdots b_{n-1} h_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ p_n a_{n-1} \cdots a_2 q_1 & p_n a_{n-1} \cdots a_3 q_2 & p_n a_{n-1} \cdots a_4 q_3 & \cdots & d_n \end{bmatrix}, \quad (9)$$

**Table 1** Sizes of generators of a block quasiseparable matrix.

	$d_k$	$q_k$	$a_k$	$p_k$	$g_k$	$b_k$	$h_k$
# of rows	$n_k$	$r_k^l$	$r_k^l$	$n_k$	$n_k$	$r_{k-1}^u$	$r_{k-1}^u$
# of cols	$n_k$	$n_k$	$r_{k-1}^l$	$r_{k-1}^l$	$r_k^u$	$r_k^u$	$n_k^u$

where parameters (called generators)  $\{d_k, q_k, a_k, p_k, g_k, b_k, h_k\}$  are matrices of sizes as in the table below.

Orders of quasiseparability  $r^l$  and  $r^u$  are maxima over the corresponding sizes of generators:

$$r^l = \max_{1 \leq k \leq n-1} r_k^l, \quad r^u = \max_{1 \leq k \leq n-1} r_k^u.$$

*Remark 1* Generators are not unique, there are infinitely many ways to represent the same quasiseparable matrix with different generators. For the relation between different sets of generators see [11].

*Remark 2* It is obvious that any scalar quasiseparable matrix can be converted to the block form by simple aggregation of its generators.

*Remark 3* Sizes  $r_k^l$  and  $r_k^u$  of generators are directly related to ranks of submatrices in the lower and upper parts, respectively. Namely, if we let  $K$  to be the block index:  $K = \sum_{i=1}^k n_i$ , then

$$\text{rank } A(K+1 : N, 1 : K) \leq r_k^l, \quad \text{rank } A(1 : K, K+1 : N) \leq r_k^u. \quad (10)$$

Moreover, for any quasiseparable matrix there exists a set of generators with sizes  $r_k^l$  and  $r_k^u$  that satisfy (10) with exact equalities (such generators are called *minimal*). For their existence and construction see [11].

One remarkable property of quasiseparable matrices is that this class is closed under inversion [9, 22]. For instance, inverse of a banded matrix is not banded but both matrices are quasiseparable. Due to the low-parametric representation in Definition 2 most of the algorithms with quasiseparable matrices have linear complexities. Table 2 lists these algorithms along with the corresponding references.

**Table 2**  $\mathcal{O}(n)$  algorithms for quasiseparable matrices

$A * v$	LU	QR	$A^{-1}$	$A * B$	$A \pm B$	$\ A\ _F$
[8]	Theorem 1	[10]	[9]	[9]	–	[11]

The ability to solve systems with quasiseparable matrices in  $\mathcal{O}(n)$  operations is essential for the purpose of this paper. One of the ways to do it is through the use of fast LU decomposition. We next derive LU decomposition

algorithm for a general block quasiseparable matrix in terms of the generators it is represented with. A restricted version of this algorithm applicable to diagonal plus semiseparable matrices (a subclass of quasiseparable matrices) was derived in [14] and some formulae of the current algorithm are similar to the ones in the inversion algorithm given in [9]. Still, to the best of our knowledge, current LU decomposition algorithm has never been published and may be useful to those who need a fast system solver for quasiseparable matrices. The complexity of the algorithm is  $\mathcal{O}(N)$  and it is valid in the *strongly regular* case (i.e. *all of the principal leading minors are non-vanishing*). First, let us note that quasiseparable structure of the original matrix implies the quasiseparable structure of  $L$  and  $U$  factors. The theorem below makes this statement precise and, in addition, relates generators of an original matrix to generators of the factors.

**Theorem 1** *Let  $A$  be a strongly regular  $N \times N$  block  $(r^l, r^u)$ -quasiseparable matrix given by generators  $\{d_k, q_k, a_k, p_k, g_k, b_k, h_k\}$  as in (9). Let  $A = LU$  be its block LU decomposition of the same block sizes. Then*

- (i) *Factors  $L$  and  $U$  are  $(r^l, 0)$ - and  $(0, r^u)$ -quasiseparable. Moreover,  $r_k^l(L) = r_k^l(A)$  and  $r_k^u(U) = r_k^u(A)$  for all  $k = 1, \dots, n-1$ .*
- (ii)  *$L$  and  $U$  are parametrized by the generators  $\{I_k, \tilde{q}_k, a_k, p_k, 0, 0, 0\}$  and  $\{\tilde{d}_k, 0, 0, 0, \tilde{g}_k, b_k, h_k\}$ , where  $I_k$  are identity matrices of sizes  $n_k \times n_k$  and new parameters  $\tilde{q}_k, \tilde{d}_k$  and  $\tilde{g}_k$  can be computed using the following algorithm:*

---

**Algorithm 1** Fast quasiseparable LU decomposition.

---

**Input:**  $d_k, q_k, a_k, p_k, g_k, b_k, h_k$   
1:  $\tilde{d}_1 = d_1, \quad \tilde{q}_1 = q_1 d_1^{-1}, \quad \tilde{g}_1 = g_1, \quad f_1 = \tilde{q}_1 \tilde{g}_1$   
2: **for**  $k = 2$  **to**  $n-1$  **do**  
3:  $\tilde{d}_k = d_k - p_k f_{k-1} h_k$   
4:  $\tilde{q}_k = (q_k - a_k f_{k-1} h_k) \tilde{d}_k^{-1}$   
5:  $\tilde{g}_k = g_k - p_k f_{k-1} b_k$   
6:  $f_k = a_k f_{k-1} b_k + \tilde{q}_k \tilde{g}_k$   
7: **end for**  
8:  $\tilde{d}_n = d_n - p_n f_{n-1} h_n$ .  
**Output:**  $\tilde{d}_k, \tilde{q}_k, \tilde{g}_k$

---

*Proof* Statement (i) of the theorem follows from statement (ii), so we only need to prove the latter part.

Denote, as usual, by  $K$  the block index:  $K = \sum_{i=1}^k n_i$  and note that quasiseparable representation (9) implies the following recurrence relation between



the blocks of  $A$ :

$$\begin{aligned}
A &= \left[ \begin{array}{c|c} A(1:K, 1:K) & G_k H_{k+1} \\ \hline P_{k+1} Q_k & A(K+1:N, K+1:N) \end{array} \right]; \\
Q_1 &= q_1, \quad Q_k = [a_k Q_{k-1} \quad q_k], \quad k = 2, \dots, n-1; \\
P_n &= p_n, \quad P_k = [p_k \quad P_{k+1} a_k], \quad k = n-1, \dots, 2; \\
G_1 &= g_1, \quad G_k = [G_{k-1} b_k \quad g_k], \quad k = 2, \dots, n-1; \\
H_n &= h_n, \quad H_k = [h_k \quad b_k H_{k+1}], \quad k = n-1, \dots, 2.
\end{aligned} \tag{11}$$

The proof will be constructed by induction. We will show that for each  $k$

$$\left[ \begin{array}{c|c} A_{11}^k & G_k H_{k+1} \\ \hline P_{k+1} Q_k & \star \end{array} \right] = \left[ \begin{array}{c|c} L_{11}^k & 0 \\ \hline P_{k+1} \tilde{Q}_k & \star \end{array} \right] \cdot \left[ \begin{array}{c|c} U_{11}^k & \tilde{G}_k H_{k+1} \\ \hline 0 & \star \end{array} \right]. \tag{12}$$

For  $k = 1$  we get from (12):

$$\begin{aligned}
d_1 &= \tilde{d}_1, & \tilde{d}_1 &= d_1, \\
P_2 Q_1 &= P_2 \tilde{Q}_1 \tilde{d}_1, & \Longleftarrow & \tilde{q}_1 = q_1 \tilde{d}_1^{-1}, \\
G_1 H_2 &= \tilde{G}_1 H_2, & \tilde{g}_1 &= g_1.
\end{aligned}$$

Let us introduce an auxiliary parameter  $f_k = \tilde{Q}_k \tilde{G}_k$ . It is easy to show by using (11) that  $f_k$  satisfies the recurrence relation

$$f_1 = \tilde{q}_1 \tilde{g}_1, \quad f_k = a_k f_{k-1} b_k + \tilde{q}_k \tilde{g}_k. \tag{13}$$

Assume that (12) holds for some fixed  $k$ , then it holds for  $k+1$  if

$$d_{k+1} = [p_{k+1} \tilde{Q}_k \quad 1] \cdot [\tilde{G}_k h_{k+1} \quad \tilde{d}_{k+1}], \tag{14}$$

$$P_{k+2} Q_{k+1} = P_{k+2} \tilde{Q}_{k+1} \cdot [\tilde{G}_k h_{k+1} \quad \tilde{d}_{k+1}], \tag{15}$$

$$g_{k+1} H_{k+2} = [p_{k+1} \tilde{Q}_k \quad 1] \cdot \tilde{G}_{k+1} H_{k+2}. \tag{16}$$

The first equality simplifies

$$d_{k+1} = p_{k+1} \tilde{Q}_k \tilde{G}_k h_{k+1} + \tilde{d}_{k+1} = p_{k+1} f_k h_{k+1} + \tilde{d}_{k+1}.$$

The second equality (15) holds if

$$\begin{aligned}
q_{k+1} &= \tilde{Q}_{k+1} \cdot [\tilde{G}_k h_{k+1} \quad \tilde{d}_{k+1}] = \\
&= [a_{k+1} \tilde{Q}_k \quad \tilde{q}_{k+1}] \cdot [\tilde{G}_k h_{k+1} \quad \tilde{d}_{k+1}] = a_{k+1} f_k h_{k+1} + \tilde{q}_{k+1} \tilde{d}_{k+1}.
\end{aligned}$$

Finally, the last equality (16) is true if

$$g_{k+1} = [p_{k+1} \tilde{Q}_k \quad 1] \cdot \tilde{G}_{k+1} = [p_{k+1} \tilde{Q}_k \quad 1] \cdot [\tilde{G}_k b_{k+1} \quad \tilde{g}_{k+1}] = p_{k+1} f_k b_{k+1} + \tilde{g}_{k+1}.$$

Matrix  $\tilde{d}_{k+1}$  is invertible because, by our assumption, matrix  $A$  is strongly regular. Hence, we conclude that (12) is true also for index  $k+1$  if generators  $\tilde{q}_{k+1}$ ,  $\tilde{d}_{k+1}$  and  $\tilde{g}_{k+1}$  are those computed in Algorithm 1. The assertion of the theorem holds by induction.  $\square$

#### 4 Multilevel quasiseparable matrices.

In the previous section we have emphasized the relation between second-order ordinary differential equations and quasiseparable matrices. Let us now look at the problem in two dimensions that extends (7):

$$\frac{\partial}{\partial x} \left( p(x, y) \frac{\partial}{\partial x} u(x, y) \right) + \frac{\partial}{\partial y} \left( q(x, y) \frac{\partial}{\partial y} u(x, y) \right) - r(x, y) u(x, y) = f(x, y)$$

for  $(x, y) \in \Omega$ , where  $\Omega = [0, 1] \times [0, 1]$  with homogeneous Dirichlet boundary conditions. The standard five-point or nine-point discretization of this problem on a  $n \times m$  uniform grid leads to a system of linear algebraic equations of the form

$$\begin{bmatrix} A_1 & B_1 & & & \\ C_1 & A_2 & B_2 & & \\ & C_2 & \ddots & \ddots & \\ & & \ddots & \ddots & B_{m-1} \\ & & & C_{m-1} & A_m \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ \vdots \\ u_m \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ \vdots \\ f_m \end{bmatrix}, \quad (17)$$

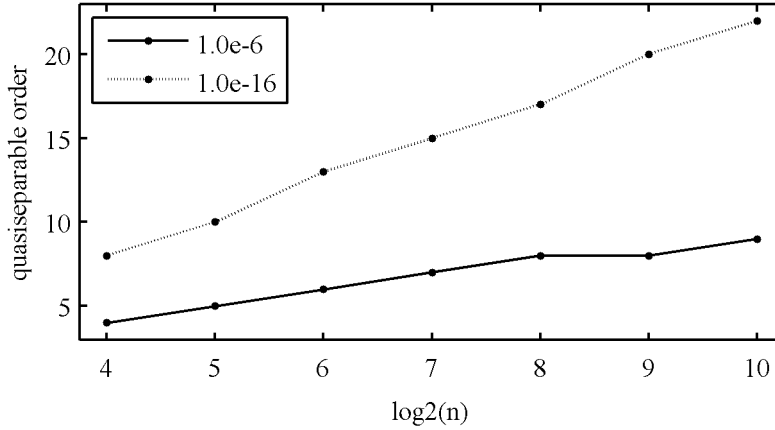
where we have assumed that  $u_i$  are the discretized unknowns along the  $i$ 'th column of the  $n \times m$  grid. In this case each of  $A_i$ ,  $B_i$ , and  $C_i$  is an  $n \times n$  matrix and the whole block matrix has size  $nm \times nm$ . Furthermore, each of the  $A_i$ ,  $B_i$ , and  $C_i$  is a tridiagonal matrix.

One way to solve system (17) is to do block LU factorization assuming that it exists. When we eliminate the block entry  $C_1$  we get in the position occupied by  $A_2$  the new block

$$S_2 = A_2 - C_1 A_1^{-1} B_1.$$

Observe that even though all the individual matrices on the right-hand side of the expression are tridiagonal,  $A_1^{-1}$  is not, and hence  $S_2$  is a dense (non-sparse) matrix. At the next step of Gaussian elimination we would use  $S_1$  as the pivot block to eliminate  $C_2$ . Now in the position occupied by the block  $A_3$  we would get the matrix  $S_3 = A_3 - C_2 S_2^{-1} B_2$ . Again, since  $S_2$  is a dense matrix, in general  $S_2^{-1}$  will be dense, and therefore  $S_3$  will also be a dense matrix. What this implies is that during LU factorization of the sparse matrix, we will produce fill-in quickly that causes us to compute inverses (and hence LU factorizations) of dense  $n \times n$  matrices. If we assume that these dense matrices have no structure, then we would need  $\mathcal{O}(n^3)$  flops for that operation alone. Therefore, it follows that one would require at least  $\mathcal{O}(mn^3)$  flops to compute block LU factorization of the system matrix.

At first glance it seems that block LU factorization is not a wise approach as it does not use any kind of fill-in minimisation reordering. However, there is a rich structure in successive Schur complements  $S_k$  that can be exploited to speed-up computations. In fact, it has been conjectured that if one looks at the off-diagonal blocks of these matrices ( $S_2$ ,  $S_3$ , etc.), then their  $\varepsilon$ -rank (number of singular values not greater than  $\varepsilon$ ) is going to be small. This conjecture has



**Fig. 1** Growth of  $\varepsilon$ -rank (quasiseparable order) with size for different  $\varepsilon$ .

been justified by the fact that, for example,  $S_k$  can be viewed approximately (especially in the limit as  $n$  becomes large) as a sub-block of the discretized Green's function of the original PDE. It is known from the theory of elliptic PDEs (see, for instance, [13]) that under some mild constraints the Green's function is smooth away from the diagonal singularity. This, in turn, implies that the numerical ranks of the off-diagonal blocks of  $S_1$  are small. There have been several recent attempts to quantify this observation. It has been proved in [5] that  $\varepsilon$ -rank of the off-diagonal blocks of Schur complements in the LU decomposition of the regular 2D-Laplacian are bounded by

$$1 + 8 \ln^4 \left( \frac{18}{\varepsilon} \right).$$

This bound is not effective for any reasonable size of the problem because of the 4'th power in the logarithm (for instance for  $\varepsilon = 1.0e-6$  it gives  $6.2e+5$ ). However, numerical experiments with Laplacian discretized by the usual five-point stencil suggest much lower bound, see Figure 1.

Careful exploitation of the low-rank structure of Schur complements reduces the complexity of block LU factorization algorithm to linear  $\mathcal{O}(mn)$ . Indeed, as it has been mentioned in the previous section, all the operations with quasiseparable matrices that are needed at each step of the algorithm have  $\mathcal{O}(n)$  complexity and there are  $m$  steps altogether. First algorithms of this type were recently developed in [25, 19]. These algorithms, although efficient in the case of simple PDEs, are not applicable to PDE-constrained optimization problems, such as the one described in Section 2. Saddle-point matrices arising there are dense block matrices with blocks themselves being discretized PDE matrices. To use block LU factorization in this case one would need to invert PDE matrices of type (17). We next show that it is possible by defining the right structured representation of inverses of PDE matrices.

**Definition 3 (Multilevel quasiseparable matrix)** Matrix  $A$  is called  $d$ -level quasiseparable if it admits the representation

$$\begin{bmatrix} D_1 & G_1 H_2 & G_1 B_2 H_3 & \cdots & G_1 B_2 \cdots B_{n-1} H_n \\ P_2 Q_1 & D_2 & G_2 H_3 & \cdots & G_2 B_3 \cdots B_{n-1} H_n \\ P_3 A_2 Q_1 & P_3 Q_2 & D_3 & \cdots & G_3 B_4 \cdots B_{n-1} H_n \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ P_n A_{n-1} \cdots A_2 Q_1 & P_n A_{n-1} \cdots A_3 Q_2 & P_n A_{n-1} \cdots A_4 Q_3 & \cdots & D_n \end{bmatrix},$$

where each of the parameters  $\{Q_k, A_k, P_k, D_k, G_k, B_k, H_k\}$  is a block matrix of blocks being  $(d-1)$ -level quasiseparable matrices. 1-level quasiseparable matrix is a usual quasiseparable matrix, see Definition 1.

Let us give some simple examples of multilevel quasiseparable matrices to justify the usefulness of the proposed structure.

*Example 1 (Block banded matrices with banded blocks)* Any banded matrix is automatically quasiseparable. Similarly, block banded matrix is 2-level quasiseparable if blocks are themselves banded matrices. As an example, consider 2D Laplacian on the square discretized by Q1 finite elements:

$$K = \begin{bmatrix} A & B & & \\ B & \ddots & \ddots & \\ & \ddots & \ddots & B \\ & & B & A \end{bmatrix}, \quad A = \frac{1}{3} \begin{bmatrix} -8 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & -8 \end{bmatrix}, \quad B = \frac{1}{3} \begin{bmatrix} 1 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 1 \end{bmatrix}. \quad (18)$$

*Example 2 (Tensor product of quasiseparable matrices)* Matrix  $A \otimes B$  is 2-level quasiseparable if  $A$  and  $B$  are 1-level quasiseparable. Hence, 2D mass matrix  $M$  on the square discretized by Q1 finite elements:

$$M = T \otimes T, \quad T = \frac{1}{6} \begin{bmatrix} 4 & 1 & & \\ 1 & \ddots & \ddots & \\ & \ddots & \ddots & 1 \\ & & 1 & 4 \end{bmatrix} \quad (19)$$

is 2-level quasiseparable. Its inverse  $M^{-1}$  is also 2-level quasiseparable due to the tensor identity  $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$  and properties of 1-level quasiseparable matrices.

Quasiseparable orders of multilevel quasiseparable matrices may differ at each level. For instance, it follows immediately from the properties of quasiseparable matrices that inverse of a 2-level quasiseparable matrix has a representation as in Definition 3 for some generators  $\{Q_k, A_k, P_k, D_k, G_k, B_k, H_k\}$ . However, there is no guarantee that entries of this generators are quasiseparable matrices of the same order as for the original matrix. 2D Laplace matrix in Example 1 is 2-level quasiseparable with generators of quasiseparable order

one but as Figure 1 illustrates, generators of LU factors (and, therefore, inverse) have orders larger than one. However, if it is true that numerical orders (ranks) do not differ much from the original orders we may say that inverses of multilevel quasiseparable matrices retain the structure approximately. This observation, in its turn, leads to algorithms of linear complexity.

To compute LU factorization of a 2-level quasiseparable matrix we may use Algorithm 1. It consists of approximately  $12n$  arithmetic operations with 1-level quasiseparable matrices. Sum or difference of rank- $r_1$  and rank- $r_2$  matrices may have rank  $r_1 + r_2$ . Similarly, any arithmetic operation (except inversion) performed on quasiseparable matrices may add their orders in the worst case. Therefore, to use multilevel quasiseparable structure efficiently we need some kind of order compression algorithm. Such algorithm was proposed in [11], we list it below for the convenience of the reader. This algorithm constitutes the essential part of the proposed method.

---

**Algorithm 2** Quasiseparable order reduction.

---

**Input:**  $q_k, a_k, p_k$  of sizes  $r_k \times n_k, r_k \times r_{k-1}$  and  $n_k \times r_{k-1}$ , respectively.

- 1: Using LQ or SVD decomposition determine matrices  $L_1$  and  $q'_1$  of sizes  $r_1 \times r'_1$  and  $r'_1 \times n_1$ , respectively, such that  $q_1 = L_1 q'_1$ , where  $q'_1 (q'_1)^* = I_{r'_1}$  and  $r'_1 = \text{rank } q_1$ .
  - 2: **for**  $k = 2$  **to**  $n - 1$  **do**
  - 3: Using LQ or SVD decomposition determine matrices  $L_k$  and  $V_k$  of sizes  $r_k \times r'_k$  and  $r'_k \times (r'_{k-1} + n_k)$ , respectively, such that  $[a_k L_{k-1} \ q_k] = L_k V_k$ , where  $V'_k (V'_k)^* = I_{r'_k}$  and  $r'_k = \text{rank}[a_k L_{k-1} \ q_k]$ .
  - 4: Split  $V_k$  into the new generators  $a'_k, q'_k$  of sizes  $r'_k \times r'_{k-1}, r'_k \times n_k$ , respectively:  
 $V_k = [a'_k \ q'_k]$ .
  - 5:  $p'_k = p_k L_{k-1}$
  - 6: **end for**
  - 7:  $p'_n = p_n L_{n-1}$
  - 8: Using QR or SVD decomposition determine matrices  $p''_n$  and  $S_{n-1}$  of sizes  $n_n \times r''_{n-1}$  and  $r''_{n-1} \times r'_{n-1}$ , respectively, such that  $p'_n = p''_n S_{n-1}$ , where  $(p''_n)^* p''_n = I_{r''_{n-1}}$  and  $r''_{n-1} = \text{rank } p'_n$ .
  - 9: **for**  $k = n - 1$  **to**  $2$  **do**
  - 10: Using QR or SVD decomposition determine matrices  $U_k$  and  $S_{k-1}$  of sizes  $(n_k + r''_k) \times r''_{k-1}$  and  $r''_{k-1} \times r'_{k-1}$ , respectively, such that  $[p'_k; S_k a'_k] = U_k S_{k-1}$ , where  $(U_k)^* U_k = I_{r''_{k-1}}$  and  $r''_{k-1} = \text{rank } p'_k$ .
  - 11: Split  $U_k$  into the new generators  $p''_k, a''_k$  of sizes  $n_k \times r''_{k-1}, r''_k \times r'_{k-1}$ , respectively:  
 $U_k = [p''_k; a''_k]$ .
  - 12:  $q''_k = S_k q'_k$
  - 13: **end for**
  - 14:  $q''_1 = S_1 q'_1$
- Output:** New generators  $q''_k, a''_k, p''_k$  of minimal possible sizes  $r''_k \times n_k, r''_k \times r'_{k-1}$  and  $n_k \times r'_{k-1}$ , respectively.
- 

In exact arithmetic, new sizes  $r''_k$  of generators match ranks of the corresponding submatrices in the quasiseparable matrix (see [11] for the proof). In floating point arithmetic we may use truncated SVD or rank-revealing LQ/QR to decrease sizes of generators to the value of the  $\varepsilon$ -rank of the corresponding

submatrix. When  $\varepsilon$ -ranks are small, complexity of Algorithm 2 is  $\mathcal{O}(n)$  and, hence, the following fact takes place.

*Complexity of algorithms listed in Table 2 with 2-level quasiseparable matrices (Definition 3) is linear in the size if quasiseparable orders of 1-level quasiseparable matrices stay small during computations.*

In the next section we will demonstrate practical properties of multi-level quasiseparable matrices approach applied to solving PDEs and PDE-constrained optimization problems.

## 5 Numerical results.

We illustrate our new method with two different examples: 2D Laplace's equation with Dirichlet boundary conditions and distributed control problem with a constraint in the form of 2D Poisson's equation. We show that the proposed method can be used either directly or as a preconditioner for an iterative method depending on the level of order reduction used in Algorithm 2. Although we consider symmetric problems only, our approach does not require symmetry and only depends on low rank properties of operators. Therefore, it can also be applied to convection–diffusion and Helmholtz problems. We do not consider 3D problems as it is not clear yet how to adapt SVD and LQ/QR decomposition used in Algorithm 2 to the block case.

Our software is written in C++ and is compiled with GCC compiler v.4.2.1. It is freely available for download at <http://www.maths.ed.ac.uk/ERG0/software.html>. All tests were done on a 2.4 GHz Intel Core 2 Duo with 4 Gb of RAM.

*Example 3* Let  $\Omega = [0, 1]^2$  and consider the problem

$$\begin{aligned} -\nabla^2 u &= 0, \text{ in } \Omega, \\ u &= \hat{u}|_{\partial\Omega}, \text{ on } \partial\Omega, \end{aligned} \tag{20}$$

where

$$\hat{u} = \begin{cases} \sin(2\pi y) & \text{if } x = 0, \\ -\sin(2\pi y) & \text{if } x = 1, \\ 0 & \text{otherwise.} \end{cases}$$

Let us introduce a square regular grid on  $[0, 1]^2$  of mesh size  $h = 1/(n+1)$  and discretize equation (20) on this grid by  $Q1$  finite elements. Then equation (20) reduces to a matrix equation with matrix  $K$  in (18). This matrix, as it was noted before, is 2-level quasiseparable and, therefore, we can apply fast LDU decomposition Algorithm 1 with fast order-reduction Algorithm 2 at each step. Maximal quasiseparable order can be chosen adaptively in SVD but we set it manually to 4 and 8 for simplicity. Larger order corresponds to better approximation and, hence, more accurate result although it makes algorithm slower. Results of the computational tests for different mesh sizes are presented in Tables 3 and 4. Note, that our solver was implemented for

unsymmetric problems and did not use symmetry of the matrix in the current example. Exploiting the symmetry would roughly halve the time and memory used.

**Table 3** Time and memory used by direct quasiseparable LDU decomposition based solver applied to the problem in Example 3, quasiseparable order (maximal off-diagonal rank) is set to **4** during computations.

$n$	LDU	mem, Mb	solve	$\frac{\ Au-f\ _2}{\ f\ _2}$
$2^{12}$	0.11	3	0.003	8.22e-05
$2^{14}$	0.51	12	0.011	1.85e-04
$2^{16}$	2.18	51	0.043	3.93e-04
$2^{18}$	9.00	210	0.169	6.91e-04
$2^{20}$	36.88	847	0.677	8.81e-04

**Table 4** Time and memory used by direct quasiseparable LDU decomposition based solver applied to the problem in Example 3, quasiseparable order (maximal off-diagonal rank) is set to **8** during computations.

$n$	LDU	mem, Mb	solve	$\frac{\ Au-f\ _2}{\ f\ _2}$
$2^{12}$	0.19	4	0.003	3.31e-09
$2^{14}$	0.91	19	0.013	6.19e-08
$2^{16}$	4.09	83	0.052	5.72e-07
$2^{18}$	17.44	342	0.209	2.33e-06
$2^{20}$	72.83	1388	0.852	5.41e-06

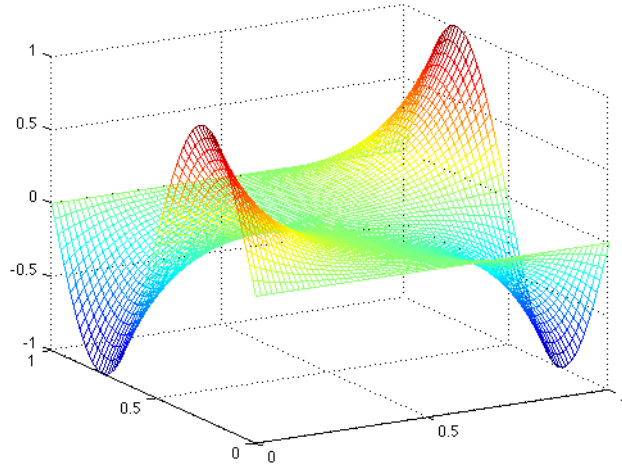
The analysis of results collected in Tables 3 and 4 reveals the linear dependence of time and memory used by the algorithm on the size of the problem. The computational experience supports our claim about *asymptotically linear complexity of algorithms with multilevel quasiseparable matrices*. Note also the linear dependence of time and the sub-linear dependence of memory on the maximal quasiseparable order used during computations.

If we truncate the order of quasiseparable matrices in the quasiseparable LDU decomposition to a very small number, the decomposition becomes less accurate. However, the algorithm becomes much faster and it is tempting to try this approximate LDU decomposition as a preconditioner in the preconditioned conjugate gradient method. Table 5 below illustrates the numerical performance of this approach. Results indicate that the PCG method preconditioned with the inexact LDU decomposition is twice as fast as more accurate LDU decomposition alone. It is enough to truncate quasiseparable order to 2–4 to force PCG to converge in less than 10 iterations.

Solution of problem in Example 3 obtained with PCG method with quasiseparable preconditioner is given in Figure 2 below.

**Table 5** Time and number of iterations (in brackets) used by PCG when applied to the problem in Example 3, preconditioner is based on approximate quasiseparable LDU decomposition with order tolerance  $r$ , system is solved to the  $1.0\text{e-}08$  accuracy.

$n$	$r$	LDU	PCG	total time
$2^{12}$	1	0.05	0.0359 (9)	0.09
	2	0.07	0.0252 (6)	0.09
$2^{14}$	1	0.21	0.212 (14)	0.42
	2	0.30	0.144 (9)	0.45
$2^{16}$	3	1.76	0.482 (7)	2.24
	4	2.18	0.282 (4)	2.47
$2^{18}$	3	7.17	2.96 (11)	10.13
	4	9.05	1.98 (7)	11.03
$2^{20}$	4	37.07	10.1 (9)	47.22
	5	45.63	8.19 (7)	53.83



**Fig. 2** Solution of problem in Example 3.

*Example 4* Let  $\Omega = [0, 1]^2$  and consider the problem

$$\begin{aligned}
 & \min_{u, f} \frac{1}{2} \|u - \hat{u}\|_{L^2(\Omega)}^2 + \beta \|f\|_{L^2(\Omega)}^2, \\
 & s.t. \quad -\nabla^2 u = f, \text{ in } \Omega, \\
 & \quad \quad u = \hat{u}|_{\partial\Omega}, \text{ on } \partial\Omega,
 \end{aligned} \tag{21}$$

where

$$\hat{u} = \begin{cases} (2x-1)^2(2y-1)^2 & \text{if } (x, y) \in [0, \frac{1}{2}]^2, \\ 0 & \text{otherwise.} \end{cases}$$



As it was already discussed in Section 2, problem (21) can be solved by the discretization by Q1 finite elements. Discretized system of equation, in this case, becomes

$$\begin{bmatrix} 2\beta M & 0 & -M \\ 0 & M & K^T \\ -M & K & 0 \end{bmatrix} \begin{bmatrix} \mathbf{f} \\ \mathbf{u} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{b} \\ \mathbf{d} \end{bmatrix}. \quad (22)$$

After we eliminate variables  $\mathbf{f}$  and  $\mathbf{u}$  in equation (22) it reduces to

$$\left( \frac{1}{2\beta} M + K M^{-1} K^T \right) \boldsymbol{\lambda} = \mathbf{y}. \quad (23)$$

This system is sometimes called *normal equation* in optimization community. Computing matrix on the left side of (23) is prohibitively expensive as it is usually very large and dense. Even more prohibitively expensive is solving the system with this matrix as it is  $\mathcal{O}(n^3)$  algorithm. One common approach is to drop either  $\frac{1}{2\beta} M$  or  $K M^{-1} K^T$  in the equation and to solve system with the remaining matrix, thus, using it as a preconditioner (see, for instance, [21]). However, this preconditioner would perform badly if none of the terms is dominant. We propose a completely different approach. Matrices  $M$  and  $K$  in (23) are multilevel quasiseparable (see Examples 1 and 2). Thus, we can compute the Schur complement matrix in the left hand side of (23) in quasiseparable arithmetic using  $\mathcal{O}(n)$  arithmetic operations if quasiseparable orders stay small during computations. The last condition is true in practice because Schur complement is itself a discretized elliptic operator similar to Laplacian. In the previous example we have shown that it is more effective to use the proposed approach as a preconditioner rather than a direct solver. Thus, our approach to solving equation (22) consists of the following steps:

1. Invert matrix  $M$  and form matrix  $S = \frac{1}{2\beta} M + K M^{-1} K^T$  in quasiseparable matrices arithmetic.
2. Compute an approximate LDU decomposition of  $S$  using Algorithm 1 and some order reduction tolerance.
3. Use PCG method to solve  $S\boldsymbol{\lambda} = \mathbf{y}$  with approximate LDU decomposition as a preconditioner.
4. Exploit computed  $\boldsymbol{\lambda}$ ,  $M^{-1}$  and the factorization (6) to find  $\mathbf{f}$  and  $\mathbf{u}$ .

We have realized the proposed approach in practice. Tables 6 and 7 gather the numerical results we have obtained while solving the problem in Example 4.

Analyzing the results presented in Tables 6 and 7 we conclude that our preconditioner is mesh-independent and, as in the case of simple PDE problem, CPU time for the overall algorithm grows linearly with the size of the problem. Control function  $f$  and obtained solution  $u$  computed with the proposed algorithm for different values of the regularization parameter  $\beta$  are presented in Figure 3.

**Table 6** Time required to construct the normal equation matrix  $S$  in the left hand side of (23) with  $\beta = 10^{-2}$ , compute its approximate LDU decomposition and solve system with it to a specified tolerance using PCG method (number of iterations is given in brackets). Quasiseparable order used in order reduction algorithm was set to **1**.

$\beta = 10^{-2}$ , tol = $10^{-4}$				
$n$	$S$	LDU( $S$ )	PCG	total time
12	0.73	1.12	0.0839 (3)	1.93
14	3.23	5.01	0.337 (3)	8.57
16	13.57	21.12	1.37 (3)	36.05
18	55.75	86.85	3.91 (2)	146.51

$\beta = 10^{-2}$ , tol = $10^{-8}$				
$n$	$S$	LDU( $S$ )	PCG	total time
12	0.71	1.11	0.13 (5)	1.95
14	3.23	5.00	0.534 (5)	8.76
16	13.58	21.11	2.17 (5)	36.86
18	55.79	86.83	8.76 (5)	151.38

**Table 7** Time required to construct the normal equation matrix  $S$  in the left hand side of (23) with  $\beta = 10^{-5}$ , compute its approximate LDU decomposition and solve system with it to a specified tolerance using PCG method (number of iterations is given in brackets). Quasiseparable order used in order reduction algorithm was set to **1**.

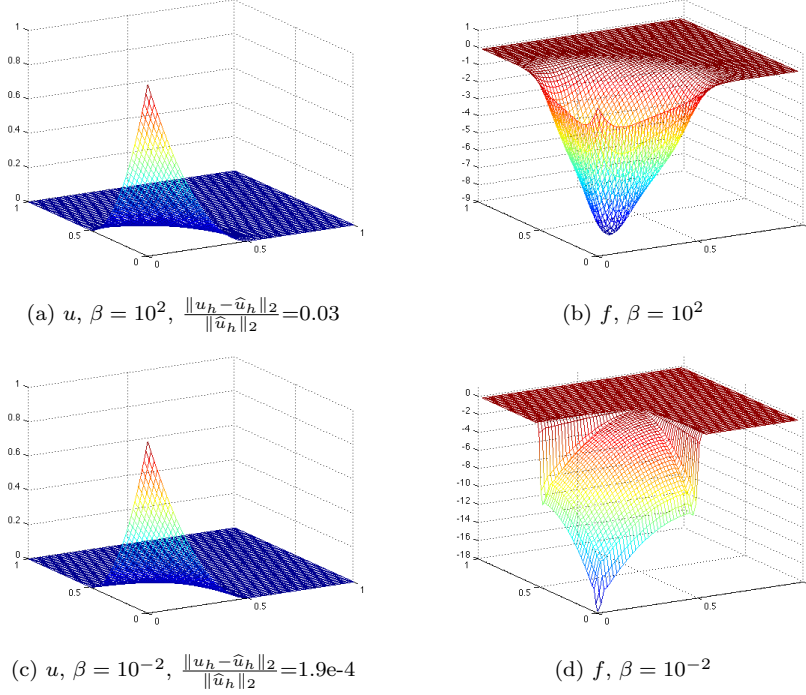
$\beta = 10^{-5}$ , tol = $10^{-4}$				
$n$	$S$	LDU( $S$ )	PCG	total time
12	0.71	1.12	0.0344 (1)	1.86
14	3.23	4.98	0.14 (1)	8.35
16	13.57	21.05	0.566 (1)	35.19
18	55.83	86.58	2.28 (1)	144.70

$\beta = 10^{-5}$ , tol = $10^{-8}$				
$n$	$S$	LDU( $S$ )	PCG	total time
12	0.71	1.11	0.058 (2)	1.88
14	3.23	4.98	0.238 (2)	8.45
16	13.57	21.05	0.966 (2)	35.59
18	55.79	86.71	3.9 (2)	146.40

## 6 Conclusion.

In this paper we have introduced a new class of rank-structured matrices called *multilevel quasiseparable*. This matrices are low-parametric in the sense that only  $\mathcal{O}(n)$  parameters are needed to store a matrix. Moreover, arithmetic operations and matrix decompositions can be performed in  $\mathcal{O}(n)$  floating-point operations. Multilevel quasiseparable matrices extend the applicability of well-known quasiseparable matrices [9] from one-dimensional to multidimensional problems. In particular, we have shown that multilevel quasiseparable structure is well-suited to the description of discretized elliptic operators in 2D.

**Fig. 3** Plots of the state,  $u$ , and the control,  $f$ , for  $\beta = 10^2$  and  $10^{-2}$ .

To demonstrate the usefulness of the new class of matrices we considered distributed control problem with a constraint in the form of a partial differential equation. Such problems were introduced by Lions and Mitter in [18]. In the course of solving them large-scale block matrices of saddle-point type arise. A straightforward way of solving these systems is to use block LU factorization. This is impractical as it requires direct inversion of large PDE matrices and further manipulations with them. However, we have shown that inverses of PDE matrices can be approximated by multilevel quasiseparable matrices with any desired accuracy and, hence, what was impossible in dense linear algebra became possible in structured linear algebra.

Development of numerical methods for solving systems of saddle-point type is an important subject of modern numerical linear algebra, see an excellent survey paper by Benzi, Golub, and Liesen [3] for details. A large amount of work has been done on developing efficient preconditioners for such systems. In the current paper we have also proposed a new efficient mesh-independent preconditioner for saddle-point systems arising in PDE-constrained optimization. Performance of the new preconditioner is studied numerically.

There are several open questions left for future research. In particular, it is not clear what the range of applicability of multilevel quasiseparable matrices is. We have only considered problems with symmetric differential operators in

our examples. However, theory does not require symmetry in the matrix and it may very well be that multilevel quasiseparable matrices are applicable to convection–diffusion and other unsymmetric operators as well. Next, robust techniques are needed to make the approach work on non-tensor grids. The biggest question though is how to extend order reduction Algorithm 2 to the block case to be able to use multilevel quasiseparable matrices in 3D and higher dimensions.

## References

1. T. Bella, Y. Eidelman, I. Gohberg, V. Olshevsky, and P. Zhlobich. Classifications of recurrence relations via subclasses of  $(H,m)$ -quasiseparable matrices. In *Numerical Linear Algebra in Signals, Systems and Control*, volume XV of *Lecture Notes in Electrical Engineering*, pages 23–53. Springer–Verlag, 2011.
2. T. Bella, V. Olshevsky, and P. Zhlobich. Signal flow graph approach to inversion of  $(H,m)$ -quasiseparable-Vandermonde matrices and new filter structures. *Linear Algebra and its Applications*, 432:2032–2051, 2010.
3. M. Benzi, G. Golub, and J. Liesen. Numerical solution of saddle point problems. *Acta Numerica*, 14:1–137, May 2005.
4. D. Bini, F. Daddi, and L. Gemignani. On the shifted QR iteration applied to companion matrices. *Electronic Transactions on Numerical Analysis*, 18:137–152, 2004.
5. S. Chandrasekaran, P. Dewilde, M. Gu, and N. Somasunderam. On the numerical rank of the off-diagonal blocks of Schur complements of discretized elliptic PDEs. *SIAM Journal on Matrix Analysis and Applications*, 31:2261–2290, 2010.
6. S. Chandrasekaran, M. Gu, and T. Pals. A Fast ULV Decomposition Solver for Hierarchically Semiseparable Representations. *SIAM Journal on Matrix Analysis and Applications*, 28(3):603–622, 2006.
7. S. Delvaux and M. Van Barel. A Givens-weight representation for rank structured matrices. *SIAM Journal on Matrix Analysis and Applications*, 29(4):1147–1170, 2007.
8. Y. Eidelman and I. Gohberg. Linear complexity inversion algorithms for a class of structured matrices. *Integral Equations and Operator Theory*, 35:28–52, 1999.
9. Y. Eidelman and I. Gohberg. On a new class of structured matrices. *Integral Equations and Operator Theory*, 34:293–324, 1999.
10. Y. Eidelman and I. Gohberg. A modification of the Dewilde-van der Veen method for inversion of finite structured matrices. *Linear Algebra and its Applications*, 343:419–450, 2002.
11. Y. Eidelman and I. Gohberg. On generators of quasiseparable finite block matrices. *Calcolo*, 42:187–214, 2005.
12. H.C. Elman, D.J. Silvester, and A.J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Oxford University Press, USA, 2005.
13. G.B. Folland. *Introduction to partial differential equations*. Princeton University Press, Princeton, NJ, 1995.
14. I. Gohberg, T. Kailath, and I. Koltracht. Linear complexity algorithms for semiseparable matrices. *Integral Equations and Operator Theory*, 8(6):780–804, 1985.
15. N.I.M. Gould, M.E. Hribar, and J. Nocedal. On the solution of equality constrained quadratic programming problems arising in optimization. *SIAM Journal on Scientific Computing*, 23(4):1376–1395, 2002.
16. W. Hackbusch. A sparse matrix arithmetic based on  $\mathcal{H}$ -matrices. Part I: Introduction to  $\mathcal{H}$ -matrices. *Computing*, 62:89–108, 1999.
17. W. Hackbusch, B. Khoromskij, and S. Sauter. On  $\mathcal{H}^2$ -matrices. In H. Bungartz, R. Hoppe, and C. Zenger, editors, *Lectures on Applied Mathematics*, pages 9–29. Springer–Verlag, Berlin, 2000.
18. J.L. Lions and S.K. Mitter. *Optimal control of systems governed by partial differential equations*, volume 396. Springer Berlin, 1971.

19. P.G. Martinsson. A fast direct solver for a class of elliptic partial differential equations. *Journal of Scientific Computing*, 38(3):316–330, 2009.
20. C.C. Paige and M.A. Saunders. Solution of sparse indefinite systems of linear equations. *SIAM Journal on Numerical Analysis*, pages 617–629, 1975.
21. T. Rees, H.S. Dollar, and A.J. Wathen. Optimal solvers for PDE-constrained optimization. *SIAM Journal on Scientific Computing*, 32(1):271–298, 2010.
22. G. Strang and T. Nguyen. The interplay of ranks of submatrices. *SIAM review*, 46(4):637–646, 2004.
23. E. Tyrtysnikov. Mosaic-skeleton approximations. *Calcolo*, 33(1):47–57, 1996.
24. R. Vandebril, M. Van Barel, and N. Mastronardi. A note on the representation and definition of semiseparable matrices. *Numerical Linear Algebra with Applications*, 12(8):839–858, 2005.
25. J. Xia, S. Chandrasekaran, M. Gu, and X.S. Li. Superfast multifrontal method for large structured linear systems of equations. *SIAM Journal on Matrix Analysis and Applications*, 31(3):1382–1411, 2009.